TECHNICAL DOCUMENT 3297
August 2015

A Survey of Algorithms to Efficiently Reconcile Sets of Infomation

Ryan Gabrys
Mark Bilinski

SSC Pacific
San Diego, CA 92152-5001

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
AUG 2015		**00-00-2015 to 00-00-2015**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
A Survey of Algorithms to Efficiently Reconcile Sets of Infomation	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **SSC Pacific,,San Diego,,CA, 92152**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
In this work, we survey existing methods that perform set reconciliation. We categorize existing algorithms into three general classes. For each class of algorithms, we provide an ideal brief description of the algorithm and then comment on its complexity and total amount of information exhange. Afterwards, areas of future work are identified.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**12**	

SSC Pacific
San Diego, California 92152-5001

K. J. Rothenhaus, CAPT, USN
Commanding Officer

C. A. Keeney
Executive Director

ADMINISTRATIVE INFORMATION

This report was prepared by the Afloat C2 Systems Engineering and Integration Branch (Code 53221) and the Adanced Information Operations (IO) Concepts Branch (Code 56150), Space and Naval Warfare Systems Center Pacific (SSC Pacific), San Diego, CA. The project work is funded by the Naval Innovative Science and Engineering Program at SSC Pacific. The project category is Basic Research.

Released by
W. Lopez, Head
Afloat C2 Systems
Engineering and Integration Branch

Under authority of
J. M. Simonetti, Head
Command & Intelligence
Systems Division

The citation of trade names and names of manufacturers in this report is not to be construed as official government endorsement or approval of commercial products or services referenced in this report.

SB

1. INTRODUCTION

An increasing need exists to maintain a Common Operational Picture (COP) between a collection of hosts within a disconnected, intermittent and low-bandwidth (DIL) Navy maritime environment. We propose the use of set reconciliation algorithms to help address this issue. Set reconciliation algorithms possess the following properties which make them promising foundations for new technology:

1. They require nearly optimal communication overhead.
2. They only use a single round of communication.
3. Many implementations possess low computational complexity.

As a first step, in this technical document, we survey the state-of-the-art with regards to the problem of set reconciliation.

The ***set reconciliation problem*** is the following: Suppose Host A and Host B each have a set of length-b binary strings, denoted $\mathcal{S}_A$ and $\mathcal{S}_B$. We briefly note that, in practice, the length-b binary strings may be the output of some hash function which is computed on the discrete data elements which constitute a Navy Command and Control (C2) data store. The problem is to determine which information must be sent between Host A and Host B so that each host can compute $\mathcal{S}_A \cup \mathcal{S}_B$ when provided a single round of communication. In other words, the protocols discussed in this document allow one exchange between A and B after which both Host A and Host B can consequently compute $\mathcal{S}_A \cup \mathcal{S}_B$. The goal in this survey is to evaluate existing set reconciliation algorithms in terms of the total amount of information exchange as well as their computational complexity. We partition the existing methods into three general classes: (1) methods based on error-correcting codes, (2) methods based on polynomial interpolation, and (3) methods based on Bloom filters.

This document is organized as follows. In Section 2., we discuss algorithms for set reconciliation based upon error-correcting codes inspired by the works [1], [8], and [11]. Section 3. describes a method for set reconciliation that leverages polynomial interpolation as in [10]. In Section 4. we describe an algorithm for set reconciliation that uses Bloom filter structures [5], [6], [7]. Lastly, in Section 5., we conclude this survey by summarizing ongoing work and identifying potential directions for future research.

2. ERROR-CORRECTING CODES

In this section, we describe an algorithm for set reconciliation that involves the direct usage of error-correcting codes. The primary advantage to this approach is that it provides nearly optimal communication overhead. The principal drawback to this approach is that the computational complexity is high. The basic idea is to represent collections of b-strings as vectors of length-b.

For a positive integer m, let $B_m : \{0, 1, 2, \ldots, 2^m - 1\} \to \mathbb{Z}_2^m$ be a function that outputs m bits which are the binary representation of an integer provided as input. Thus, $B_3(2) \to (0, 1, 0)$ and $B_3(1) = (0, 0, 1)$, for instance. Clearly, B_m is invertible. Let $M : \mathbb{Z}_2^b \to \mathbb{Z}_2^{2^b}$. Then, for any input $\boldsymbol{x}$, $M(\boldsymbol{x}) = (y_0, y_1, \ldots, y_{2^b-1})$ is a vector, with $y_{B_b^{-1}(\boldsymbol{x})} = 1$ and $y_i = 0$ otherwise. Under this representation, we represent our sets $\mathcal{S}_L$ ($L \in \{A, B\}$) by vectors $\boldsymbol{v}_L$ ($L \in \{A, B\}$), where

$$\boldsymbol{v}_L = \sum_{\boldsymbol{x} \in \mathcal{S}_L} M(\boldsymbol{x}).$$

Suppose $H \in \mathbb{Z}_2^{r \times 2^b}$ is a parity check matrix for a code that corrects up to $2d+1$ errors and suppose $|(\mathcal{S}_A \setminus \mathcal{S}_B) \cup (\mathcal{S}_B \setminus \mathcal{S}_A)| \leq d$. Then, using H we compute

$$\boldsymbol{h}_L = H \cdot \boldsymbol{v}_L, \tag{1}$$

for $L \in \{A, B\}$. Equation (1) represents the encoding which takes place on both Host A and Host B. Notice that one potential disadvantage with this setup is that $\boldsymbol{v}_A, \boldsymbol{v}_B$ are exponential in the parameter b so that the size of the vectors $\boldsymbol{v}_A, \boldsymbol{v}_B$ could be quite large, making the matrix multiplication in Equation (1) an expensive operation.

The next step is for Host A to send $\boldsymbol{h}_A$ to Host B and Host B to send $\boldsymbol{h}_B$ to Host A. In the following, we focus on the computation of $\mathcal{S}_A \cup \mathcal{S}_B$ on Host B. The logic on Host A is identical. Since Host B now has $\boldsymbol{h}_A, \boldsymbol{h}_B$, Host B can compute

$$\begin{aligned}\boldsymbol{h}_A + \boldsymbol{h}_B = H \cdot \boldsymbol{v}_A + H \cdot \boldsymbol{v}_B =& H \cdot \left(\sum_{\boldsymbol{x} \in \mathcal{S}_A} M(\boldsymbol{x}) + \sum_{\boldsymbol{x} \in \mathcal{S}_B} M(\boldsymbol{x}) \right) \\ &= H \cdot \sum_{\boldsymbol{x} \in (\mathcal{S}_A \setminus \mathcal{S}_B) \cup (\mathcal{S}_B \setminus \mathcal{S}_A)} M(\boldsymbol{x}) \\ &= H \cdot M(\boldsymbol{e}),\end{aligned}$$

where $M(\boldsymbol{e}) = \sum_{\boldsymbol{x} \in (\mathcal{S}_A \setminus \mathcal{S}_B) \cup (\mathcal{S}_B \setminus \mathcal{S}_A)} M(\boldsymbol{x})$. For a vector $\boldsymbol{v} \in \mathbb{Z}_2^b$, let $wt(\boldsymbol{v})$ denote the number of non-zero elements in $\boldsymbol{v}$. Thus, by assumption, $wt(M(\boldsymbol{e})) \leq d$ and, since H is the parity check matrix for a code with minimum distance $2d + 1$, we can uniquely determine $\sum_{\boldsymbol{x} \in (\mathcal{S}_A \setminus \mathcal{S}_B) \cup (\mathcal{S}_B \setminus \mathcal{S}_A)} M(\boldsymbol{x})$ from which the set $(\mathcal{S}_A \setminus \mathcal{S}_B) \cup (\mathcal{S}_B \setminus \mathcal{S}_A)$ can be reconstructed.

Let $E(b, d)$ denote the total amount of information exchange. It can be shown [12] that the total amount of information that is required to be transmitted from Host A to Host B (and likewise from Host B to Host A) is at most

$$E(b, d) \leq db.$$

The following theorem summarizes the discussion from this section.

Theorem 1. *There exists a set reconciliation protocol that requires db bits of information exchange with encoding complexity $O(d \cdot 2^b)$ and decoding complexity $O(d \cdot 2^b)$.*

3. POLYNOMIAL INTERPOLATION

In this section, we describe the set reconciliation approach from [10] that leverages polynomial interpolation. We first describe the approach along with some of its properties. To describe the polynomial interpolation method, we first introduce the concept of the ***characteristic polynomial*** ([2], [9]) which will serve to represent the sets $\mathcal{S}_A, \mathcal{S}_B$ on Host A and Host B, respectively. For a set $\mathcal{S} = \{X_1, X_2, \ldots, X_n\} \subseteq GF(q)$ where $q \geq 2^b$, we define the characteristic polynomial of $\mathcal{S}$ as

$$\chi_{\mathcal{S}}(z) = (z - X_1)(z - X_2) \cdots (z - X_n).$$

Host A and Host B first agree upon d evaluation points, denoted $\{P_1, P_2, \ldots, P_d\} \subseteq GF(q)$, and we assume $|(\mathcal{S}_A \setminus \mathcal{S}_B) \cup (\mathcal{S}_B \setminus \mathcal{S}_A)| \leq d$. Host A transmits the result of evaluating $\chi_{\mathcal{S}_A}(z)$ at each of the d evaluation points along with the number $|\mathcal{S}_A|$ to Host B. Host B transmits the result of evaluating $\chi_{\mathcal{S}_B}(z)$ at each of the d evaluation points along with $|\mathcal{S}_B|$ to Host A. Similar to before, we describe the process of determining $\mathcal{S}_A \cup \mathcal{S}_B$ on Host B since the process on Host A is the same.

At this point, Host B has the following information:

1. $|\mathcal{S}_A|, |\mathcal{S}_B|$, and
2. $\{\chi_{\mathcal{S}_A}(P_1), \chi_{\mathcal{S}_A}(P_2), \ldots, \chi_{\mathcal{S}_A}(P_d)\}, \{\chi_{\mathcal{S}_B}(P_1), \chi_{\mathcal{S}_B}(P_2), \ldots, \chi_{\mathcal{S}_B}(P_d)\}$.

On Host B, we seek to recover the polynomial

$$\frac{\chi_{\mathcal{S}_A}(z)}{\chi_{\mathcal{S}_B}(z)} = \frac{\chi_{\mathcal{S}_A \setminus \mathcal{S}_B}(z)}{\chi_{\mathcal{S}_B \setminus \mathcal{S}_A}(z)},$$

from which the set $\mathcal{S}_A \cup \mathcal{S}_B$ can be easily determined. Without loss of generality, suppose $|\mathcal{S}_A| \geq |\mathcal{S}_B|$ and denote $d_A = |\mathcal{S}_A \setminus \mathcal{S}_B|$, $d_B = |\mathcal{S}_B \setminus \mathcal{S}_A|$. Let $\delta = |\mathcal{S}_A| - |\mathcal{S}_B|$. Then, $d_A \leq \lfloor \frac{d+\delta}{2} \rfloor$ and $d_B \leq \lfloor \frac{d-\delta}{2} \rfloor$. Denote

$$\frac{\chi_{\mathcal{S}_A \setminus \mathcal{S}_B}(z)}{\chi_{\mathcal{S}_B \setminus \mathcal{S}_A}(z)} = \frac{q(z)}{r(z)} = \frac{z^{d_A} + q_{d_A-1} z^{d_A-1} + \cdots + q_0}{z^{d_B} + r_{d_B-1} z^{d_B-1} + \cdots + r_0}.$$

Then, for $i \in \{1, 2, \ldots, d\}$, we have that

$$\frac{\chi_{\mathcal{S}_A \setminus \mathcal{S}_B}(P_i)}{\chi_{\mathcal{S}_B \setminus \mathcal{S}_A}(P_i)} = \frac{P_i^{d_A} + q_{d_A-1} P_i^{d_A-1} + \cdots + q_0}{P_i^{d_B} + r_{d_B-1} P_i^{d_B-1} + \cdots + r_0}. \tag{2}$$

If $F_i = \frac{\chi_{\mathcal{S}_A \setminus \mathcal{S}_B}(P_i)}{\chi_{\mathcal{S}_B \setminus \mathcal{S}_A}(P_i)}$, then for $i \in \{1, 2, \ldots, d\}$ we can rewrite Equation (2) as

$$P_i^{d_A} + q_{d_A-1} P_i^{d_A-1} + \cdots + q_0 = F_i \left(P_i^{d_B} + r_{d_B-1} P_i^{d_B-1} + \cdots + r_0 \right). \tag{3}$$

Since it can be shown the equations from Equation (3) are linearly independent [13], we can uniquely determine $\frac{\chi_{\mathcal{S}_A \setminus \mathcal{S}_B}(P_i)}{\chi_{\mathcal{S}_B \setminus \mathcal{S}_A}(P_i)}$ from the previous derivations.

Notice that the polynomial interpolation method removed the requirement that operations are performed over binary vectors of length 2^b. However, the encoding procedure does require operations over a field of size $q > 2^b$ so that it is unclear whether the polynomial interpolation method provides any meaningful advantages in terms of encoding complexity over the approach which uses error-correcting codes. The communication overhead is the same as the method from the previous section ($O(db)$), while the decoding complexity is $O(d^3)$. Notice that if $d^2 << 2^b$, then the polynomial interpolation method may offer a substantial improvement in decoding complexity.

The next theorem summarizes the discussion from this section.

Theorem 2. *There exists a set reconciliation protocol that requires db bits of information exchange with decoding complexity $O(d^3)$.*

4. BLOOM FILTER

In this section, we consider a slightly different approach than those taken in the previous two sections. Recall that in the previous two sections, the protocols discussed guarantee recovery of $\mathcal{S}_A \cup \mathcal{S}_B$ whenever $|(\mathcal{S}_A \setminus \mathcal{S}_B) \cup (\mathcal{S}_B \setminus \mathcal{S}_A)| \leq d$. In contrast, the Bloom filter approach allows recovery of $\mathcal{S}_A \cup \mathcal{S}_B$ with ***high probability*** whenever $|(\mathcal{S}_A \setminus \mathcal{S}_B) \cup (\mathcal{S}_B \setminus \mathcal{S}_A)| \leq d$. For the remainder of this section, we discuss a variation of the protocol from [5] which uses the invertible Bloom filter first discussed in [6]. Similar ideas were also used in [3] and [7].

On each host, a special type of Bloom filter, known as an invertible Bloom filter (IBF) is created. These hosts first agree on two hash functions H and H_k for some positive integer k. The IBF is a collection of n cells. For each b-string, H_k hashes it into k cells of the IBF. Each cell then contains three fields:

1. $idSum$: XOR of all b-strings that have hashed into the cell.
2. $hashSum$: XOR of H of all b-strings that have hashed into the cell.
3. $count$: an integer counting the number of times the cell has been hashed to.

The idea will be for Host A to compute an IBF, denoted $B_{\mathcal{S}_A}$, and for Host B to compute an IBF, denoted $B_{\mathcal{S}_B}$. Afterwards, A and B will exchange IBFs and, from the IBFs, Host A and B will determine $\mathcal{S}_A \cup \mathcal{S}_B$. We assume that we have the hash functions $h_1, \ldots, h_k$ where for $i \in \{1, \ldots, k\}$, $h_i : \mathbb{Z}_2^b \to \mathbb{Z}_n$. $H = h_0$ and $H_k = (h_1, ..., h_k)$. The computation of $B_{\mathcal{S}_A}$ is detailed in Algorithm 1 when $\mathcal{S} = \mathcal{S}_A$ and similarly $B_{\mathcal{S}_B}$ is computed from Algorithm 1 when $\mathcal{S} = \mathcal{S}_B$. It is assumed that at the start of the algorithm the IBFs $B_{\mathcal{S}_A}, B_{\mathcal{S}_B}$ are empty. Let $\oplus$ denote the XOR operation so that $(1, 0, 1, 0) \otimes (1, 1, 0, 0) = (0, 1, 1, 0)$.

Algorithm 1: IBF Encode

input : The set $\mathcal{S} \subseteq GF(2^b)$
output: The IBF $B_{\mathcal{S}}$

for *every* $\boldsymbol{x} \in \mathcal{S}$ **do**
  for $i = 1 : k$ **do**
    $B_{\mathcal{S}}[h_i(\boldsymbol{x})]_{.idSum} = B_{\mathcal{S}}[h_i(\boldsymbol{x})]_{.idSum} \oplus x$;
    $B_{\mathcal{S}}[h_i(\boldsymbol{x})]_{.hashSum} = B_{\mathcal{S}}[h_i(\boldsymbol{x})]_{.hashSum} \oplus H(x)$;
    $B_{\mathcal{S}}[h_i(\boldsymbol{x})]_{.count} = B_{\mathcal{S}}[h_i(\boldsymbol{x})]_{.count} + 1$;
  end
end

After Host A computes $B_{\mathcal{S}_A}$ and Host B computes $B_{\mathcal{S}_B}$, Host A sends $B_{\mathcal{S}_A}$ to Host B and similarly Host B sends $B_{\mathcal{S}_B}$ to Host A. The decoding procedure detailed in Algorithm 2 is performed on Both Host A and Host B. The operation $B_{\mathcal{S}_B}.j - B_{\mathcal{S}_A}.j$ returns an array where the value in the i-th cell of the resulting array is equal to $B_{\mathcal{S}_B}[i].j - B_{\mathcal{S}_A}[i].j$ for $i \in \{1, 2, \ldots, n\}$ and where j is one of the three fields $idSum$, $hashSum$, $count$. Similarly the operation $B_{\mathcal{S}_B}.j \oplus B_{\mathcal{S}_A}.j$ returns an array where the value in the i-th cell of the resulting array is equal to $B_{\mathcal{S}_B}[i].j \oplus B_{\mathcal{S}_A}[i].j$ for $i \in \{1, 2, \ldots, n\}$ and where j is one of the three fields $idSum$, $hashSum$, $count$.

Algorithm 2: IBF Decode

input : $B_{\mathcal{S}_A}, B_{\mathcal{S}_B}$
output: An estimate $\mathcal{F}$ of $(\mathcal{S}_A \setminus \mathcal{S}_B) \cup (\mathcal{S}_B \setminus \mathcal{S}_A)$

$\mathcal{F} = \emptyset$;
$\ell = 1$;
$B_{.idSum} = B_{\mathcal{S}_B.idSum} \oplus B_{\mathcal{S}_A.idSum}$;
$B_{.hashSum} = B_{\mathcal{S}_B.hashSum} \oplus B_{\mathcal{S}_A.hashSum}$;
$B_{.count} = B_{\mathcal{S}_B.count} - B_{\mathcal{S}_A.count}$;
while $\ell \leq n$ **do**
  if $B[\ell]_{.count} = \pm 1 \wedge B[\ell]_{.hashSum} = H(B[\ell]_{.idSum})$ **then**
    $\boldsymbol{x} = B[\ell]_{.idSum}$;
    $\mathcal{F} = \mathcal{F} \cup \boldsymbol{x}$;
    for $i = 1 : k$ **do**
      $B[h_i(\boldsymbol{x})]_{.idSum} = B[h_i(\boldsymbol{x})]_{.idSum} \oplus \boldsymbol{x}$;
      $B[h_i(\boldsymbol{x})]_{.hashSum} = B[h_i(\boldsymbol{x})]_{.hashSum} \oplus H(\boldsymbol{x})$;
      $B[h_i(\boldsymbol{x})]_{.count} = B[h_i(\boldsymbol{x})]_{.count} - B[\ell]_{.count}$;
    end
    $\ell = 0$;
  end
  $\ell = \ell + 1$;
end
If $B_{.count}$ does not contain all-zeros, then a decoding error has occurred.

When the number of cells in the IBF satisfies $n \geq d(k+1)$, we have the following theorem which follows in a straightforward manner from the ideas in [5].

Theorem 3. *There exists a set reconciliation protocol that requires $O(db)$ bits of information exchange with encoding complexity $O(\max\{|\mathcal{S}_A|, |\mathcal{S}_B|\})$ and decoding complexity $O(d)$ that computes $\mathcal{S}_A \cup \mathcal{S}_B$ with probability at least $1 - O(d^{-k})$ when $|(\mathcal{S}_A \setminus \mathcal{S}_B) \cup (\mathcal{S}_B \setminus \mathcal{S}_A)| \leq d$.*

5. APPLICATIONS AND FUTURE WORK

In this section, we comment on one possible Navy system, known as Maritime Tactical Command and Control (MTC2), which could benefit from the use of set reconciliation algorithms. Afterwards, we consider directions for future work.

MTC2 is the follow-on to Global Command and Control System-Maritime (GCCS-M) and provides capabilities that include situational readiness and planning for Navy Tactical environments. One of the core components of MTC2 is the data layer which abstracts the implementation of the underlying data store from MTC2 applications. In particular, the MTC2 data layer provides a RESTful interface to a schemaless database. All documents stored within the data layer have the format shown in Figure 1.

document structure

```
{"_id" : { "$oid" : "<mongo uuid>"} ,
 "@context" : {
                 "mtc2": "https://mil.navy.mtc2/1.0",
                 "@base": null,
                 "@vocab": "mtc2:"
              },
 "@id" : "<java uuid>",
 "@type" : ["<some type>"],
 "versionVector": {<version vector}, // Version Vector - see below
 "createdDate": <date>,
 "createdBy": <userId>,
 "metadata":[{"name":"<some name>","value":"<some value>"}, ...]} // Application
defined
 "data" : {Current Document},
 "isDeleted" : <true | false>,
 "isHistory" : <true | false>,
 }
```

Figure 1. Document Structure.

Suppose that two data stores, denoted Host A and Host B, within MTC2 want to synchronize their information. One simple approach under this setup could be the following. First, suppose a hash field is added to each document (where the hash is computed using the data content within each document). Host A determines which hashes it has that Host B does not have and similarly Host B determines the hashes it has that Host A does not have. As mentioned previously, if a set reconciliation algorithm is used to determine the hash differences, then only a single round of communication is required and the amount of information exchanged between Host A and Host B is nearly optimal. Host A then sends to Host B the documents corresponding to the hashes Host A has that Host B does not and similarly Host B sends a set of documents to Host A.

As a concrete example of the benefit of using these algorithms suppose Host A has a set of documents which are 1 KB each in size and similarly Host B has a set of documents each of size 1 KB. Furthermore, suppose we use a 32-bit Cyclic Redundancy Check (CRC32) hash so that the size of each hash is 32 bits. Suppose Host A has 10 documents that Host B does not have and similarly Host B has 10 documents that Host A does not have. Then, using the approach described in the previous paragraph would require the transmission of 164480 bits. If Host A knew all the documents Host B had and Host B knew all the documents Host A had, then 163840 bits of information would be exchanged so that under this setup, our approach would be nearly optimal in terms of the total amount of information exchange. Furthermore, notice that only two rounds of communication would be required (one round for the hashes and another to transmit the documents). Such approaches may be suitable for degraded naval networks where bandwidth may be a scarce resource.

Some directions for future work include the following:

1. Set reconciliation algorithms for data elements that are related.
2. New reconciliation algorithms with nearly optimal information exchange that possess less computational complexity than polynomial interpolation.

3. Algorithms with security constraints.

The first item above is motivated by the setup where a host makes minor changes to a document between synchronization rounds. For instance suppose Host A and B initially have a single, identical track. Suppose Host A updates the track location information and leaves the rest of the track the same as before. We would like to have an algorithm for synchronizing A and B that only transmits the updates that Host A made to the track rather than the entire track itself. Such a method has the potential to further reduce the required throughput of existing algorithms. Some preliminary work towards the development of such algorithms has started in [4].

Recall that the approach outlined in Section 4. reduced the computational complexity, but the amount of communication overhead was then increased by a constant factor. This constant factor of additional communication may be prohibitively expensive in DIL environments. Consequently, the second direction enumerated above proposes to reduce the communication overhead required by approaches that use Bloom filters at the cost of potentially increasing the computational complexity of the algorithm from $O(d)$ to $O(d \log d)$.

The third item identified for future work refers to the scenario where a collection of hosts communicating together have different security privileges. Therefore, it may be desirable to leverage the structure of the set reconciliation transmission schemes to enhance the privacy of the information exchanged.

REFERENCES

[1] K.A.S. Abdel-Ghaffar and A.E. Abbadi, 1994. “An Optimal Strategy for Comparing File Copies,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 5, no. 1, pp. 87–93 (January).

[2] M. Blum and S. Kannan. 1995. “Designing Programs That Check Their Work,” *Journal of the ACM*, vol. 42, no. 1, pp. 269–291(January).

[3] R. Gabrys and D. Coker. 2015. “Set Reconciliation in Two Rounds of Communication,” International Command and Control Research and Technology Symposium (ICCRTS). June 16–19, Annapolis, MD.

[4] R. Gabrys and F. Farnoud. 2015. “Reconciling Similar Sets of Data,” IEEE International Symposium on Information Theory (ISIT), June 14–19, Hong Kong,

[5] D. Eppstein, M. Goodrich, F. Uyeda, and G. Varghese, 2011. “What’s the Difference? Efficient Set Reconciliation without Prior Context,” *Proceedings of the ACM SIGCOM 20112 Conference* (pp. 218–229). Aug. 15–19, Toronto, Canada.

[6] M. T. Goodrich and M. Mitzenmacher, 2011. “Invertible Bloom Lookup Tables,” ArXiv e-prints.

[7] D. Guo and M. Li. 2013. “Set Reconciliation via Counting Bloom Filters,” *IEEE Transactions on Knowledge and Data Eng*ineering, vol. 25, no. 10, pp. 2367–2380 (October).

[8] M. Karpovsky, L. Levitin, and A. Trachtenberg. 2001. “Data Verification and Reconciliation with Generalized Error-control Codes,” 39th Annual Allerton Conference on Communication, Control, and Computing, October 3–5, Monticello, IL.

[9] R. J. Lipton. 1990. “Efficient Checking of Computations.” *Proceedings of the 7th Annual Symposium on Theoretical Aspects of Computer Science (STACS)* (pp. 207–215). February 22–24, Rouen, France.

[10] Y. Minsky, A. Trachtenberg, and R. Zippel. 2003. “Set Reconciliation with Nearly Optimal Communication Complexity,” *IEEE Transactions Information Theory*, vol. 49, no. 9, pp. 2213–2218
(September).

[11] A. Orlitsky. 2006. “Communication Issues in Distributed Computing.” Doctoral thesis. Stanford University
Electrical Engineering Department. Stanford, CA.

[12] R. Roth. 2006, *Introduction to Coding Theory*. Cambridge University Press, New York, NY.

[13] R. E. Zippel. 1993. *Effective Polynomial Computation*. Kluwer Academic Press, Boston, MA.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-01-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
August 2014	Final	

4. TITLE AND SUBTITLE

A Surve of Algoriths to Efficiently Reconcile Sets of Iformation

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHORS

Ryan Gabrys
Mark Bilinski

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

SSC Pacific, 53560 Hull Street, San Diego, CA 92152–5001

8. PERFORMING ORGANIZATION REPORT NUMBER

TD 3297

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Innovative Science and Engineering Program (Basic Research)
53560 Hull Street
San Diego, CA 92152–5001

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this work, we survey existing methods that perform set reconciliation. We categorize existing algorithms into three general classes. For each class of algorithms, we provide an ideal brief description of the algorithm and then comment on its complexity and total amount of information exhange. Afterwards, areas of future work are identified.

15. SUBJECT TERMS

Mission Area: Command and Control

algorithms	polynomial interpolation	common operational picture
Error-correcting codes	Bloom filter	disconnected, intermittet and lowbandith

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Ryan Gabrys
U	U	U	U	12	**19B. TELEPHONE NUMBER** *(Include area code)* (619) 553-1900

INITIAL DISTRIBUTION

84300	Library	(2)
85300	Archive/Stock	(1)
53221	R. Gabrys	(1)
56150	M. Bilinski	(1)

Defense Technical Information Center
Fort Belvoir, VA 22060–6218 (1)

Approved for public release.

SSC Pacific
San Diego, CA 92152-5001